

APACHE - 2 WEB SERVER

**Course
+ Exercises**

(English)

Web Server Course Overview

1 - Proper server partitioning for home pages and logs

- /var, /home, /srv

2 - Components needed:

- Web server (Apache, thttpd, ...)
 - Normal: http://....
 - Secure SSL: https://....
- FTP Server (vsftpd, pureftpd, proftpd, in.ftpd, wu-ftpd)
 - Users login for Home Pages Upload
 - Anonymous for extra download items from Directory

3 - Web Server

Theory

- What is a Web Server (thttpd, MSInformation Server, Netscape, AOL Server etc.
- TCP/IP and the HTTP/HTTPS Protocol
- What is URL and URI
- **Name to IP resolving principle without Proxy**
- **Name to IP resolving principle with Proxy**
- Browsers
- Differences between: thttpd, Apache 1.3 and Apache 2.
- How to get Apache
- Preparation for Exercises:
- Start/Stop of Apache
- First Exercise with HTTP Protocol
- Watching the http protocol with Ethereal
- Configuration files
- Where is what
- User and group setting for client processes
- Modules : Roles and their Directives
- Apache Server status
- Documentation
- Important core directives

Practice

- Enabling server-status and server-info modules
- Preparation for Exercises
- Alias Directive
- Containers
- Options FollowSymlinks
- DirectoryIndex
- Allow - Deny
- Public_html
- Authentication

- INDEXES
- AllowOverride
- Virtual Hosts
- CGI
- SUDO
- Verify that the PHP module works
- Creating web access reports with Webalizer
- Secure HTTP: HTTPS configuration:
- Creating a temporary self signed certificate for SSL:
- Secure Webmail Installation (Squirrelmail)
- Picture Galery project: IDS - CGI
- httpd2-prefork command Options
- Controlling the started processes and clients on Apache
- Apache in Debian based distributions
- Writing Apache configuration scripts

4 - FTP server: Installing and configuring vsftpd

- For Web Pages upload
- For file downloads

5 - Installing LAMP - Linux-Apache-MySQL-PHP

- Installing MySQL and PHPMyAdmin

6 - Getting and Installing WAMP-Windows-Apache-MySQL-PHP

7 - Getting and installing XAMP for MAC OS (Intel Only)

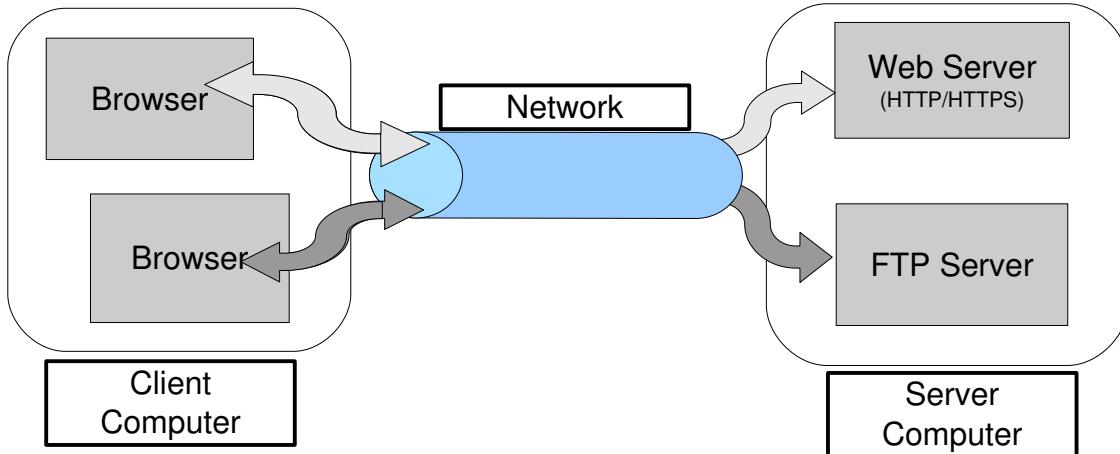
Table of Contents

What is a Web Server	6
TCP/IP and the HTTP/HTTPS Protocol.....	6
What is URL and URI	6
Browsers.....	6
Differences between: httpd, Apache 1.3 and Apache 2.....	6
How to get Apache.....	7
Apache for Linux.....	7
Apache for Windows.....	7
Apache for MAC OS X.....	7
Installing Apache in SuSE.....	8
Start/Stop of Apache.....	8
First Exercise with HTTP Protocol.....	8
Watching the http protocol with Ethereal.....	8
Configuration files.....	8
Where is what.....	8
User and group settings for clients processes.....	8
Modules.....	8
Apache Server status.....	8
Documentation.....	9
Important core directives.....	9
Preparation for Exercises.....	9
Some troubleshooting commands.....	9
Testing the web server's home pages.....	9
ALIAS.....	9
Containers.....	10
Options FollowSymlinks.....	10
DirectoryIndex.....	10
Allow - Deny.....	11
Public_html.....	12
Authentication.....	12
INDEXES.....	13
AllowOverride.....	15
Virtual Hosts.....	15
Writing scripts for Apache administration.....	16
CGI.....	17
cgitest.cgi.....	18
SUDO.....	18
Verify that the PHP module works.....	19
Creating web access reports with Webalizer.....	19
Secure HTTP: HTTPS configuration:.....	20
Creating a temporary self signed certificate for SSL.....	21
Secure Webmail Installation.....	21
IDS Images Galerie - CGI Installation.....	22
httpd2-prefork command Options.....	23
Controlling the started processes and clients on Apache.....	23

Apache in Debian based distributions.....	24
Writing Apache configuration scripts.....	24
Options Multiviews.....	24
LAMP - Linux-Apache-MySQL-PHP.....	25
WAMP-Windows-Apache-MySQL-PHP.....	27
Installing Apache2, PHP4 and MySQL in Windows2000/XP.....	27
Manual installation for the brave!.....	27
INSTALLING APACHE2.....	27
INSTALLING MYSQL.....	27
INSTALLING PHP4.....	27
INSTALLING phpMyAdmin.....	27
CONFIGURATION.....	28
MY.CNF (Example from XAMPP).....	29

Apache2 (SuSE 10.1)

- **What is a Web Server** (thttpd, MSInformation Server, Netscape, AOL Server etc.)



- **TCP/IP and the HTTP/HTTPS Protocol**

- TCP Transport the package
- HTTP is the language of the message transported
- GET and POST are the two most used commands by browsers to request a resource. (web page, image, pdf document etc.)

- **What is URL and URI**

Uniform Resource Locator

Uniform Resource Identifier

String identifying a resource by name and possibly including location.

example of URL: http:// www.elop.de /bilder/kopf1.jpg

1 2 3

1: Protocol

2: ServerAddress

3: Location and Resource(**URI**)

- **Browsers**

Linux:

- Mozilla, Opera, Firefox, Netscape, Konqueror, Galeon, Browsex, etc

Windows:

- MSIE, Mozilla, Opera, Firefox, Netscape

MAC OS X:

- Safari, Opera, Firefox, Camino

Text:

- lynx, w3m

- **Differences between: thttpd, Apache 1.3 and Apache 2.**

- thttpd : Included in the kernel, just need to activate with some tools
- Apache 1.3

- Forking based server

- One central core unit and dynamic modules

- Apache 2.x

- Contains the Core, 2 sub-core (servers) and dynamic modules

- Core : Central part of the server

- Subcores:

- apache2-prefork
 - Copies itself as a separate process (forking) to serve each new client.
 - Advantages: More modules are compatible with it.
 - Disadvantages: Slower than the worker.
Needs more RAM
- apache2-worker
 - Starts a processes like in prefork but each process can serve multiple clients instead of only one(as seen with prefork). Each new process will in turn start sub-processes called 'Threads' to serve each new client.
 - Advantages: Faster than the prefork
More memory efficient
 - Disadvantages: Less modules are compatible to it.
- apache2-perchild
 - Similar to 'worker' except that each Virtual Host is assigned a separate process which in turns uses separate 'Threads' to serve the clients.
This allows also to assign a separate user and group per process which eliminates the need of SUExec.
 - Advantages: Flexible: Can assign different users per Virtual Hosts. Memory efficient.
 - Disadvantages: Less modules are compatible to it.
- Dynamic Modules
 - Pieces of software that can attach themselves to the core of Apache to add functionality.
 - Note:** Most modules come with set of Directives that can be used in the Apache configuration file to control the behavior of the module.

- **How to get Apache**

- Apache for Linux

- Most distributions provide already compiled Apache.
- Source code of Apache can be found at: <http://www.apache.org>

- Apache for Windows can also be downloaded from:

<http://www.apachefriends.org/en/xampp-windows.html>

The package contains:

Apache HTTPD 2.2.3, MySQL 5.0.27, PHP 5.2.0 + 4.4.4 + PEAR + Switch, MiniPerl 5.8.7, OpenSSL 0.9.8d, phpMyAdmin 2.9.1.1, XAMPP Control Panel 2.3, Webalizer 2.01-10, Mercury Mail Transport System für Win32 und NetWare Systems v4.01a, FileZilla FTP Server 0.9.20, SQLite 2.8.15, ADODB 4.93, Zend Optimizer 3.0.2, XAMPP Security. For Windows 98, 2000, XP.

- Apache for MAC OS X can also be downloaded from:

<http://www.apachefriends.org/en/xampp-macosx.html>

The package contains:

Apache 2.2.3, MySQL 5.0.27, PHP 4.4.4, PHP 5.2.0, Perl 5.8.8, ProFTPD 1.3.0a, phpMyAdmin 2.9.1, OpenSSL 0.9.8d, GD 2.0.33, Freetype 2.1.10, libjpeg 6b, libpng 1.2.12, libungif-4.1.4, zlib 1.2.3, expat 2.0.0, Ming 0.3, Webalizer 2.01-10, pdf class 009e, mod_perl 2.0.2, SQLite 3.3.7, phpSQLiteAdmin 0.2, libiconv-1.9.2, gdbm-1.8.3, libxml-2.6.26, libxml-1.1.17, openldap-2.3.27, imap-2004g, gettext-0.15, libmcrypt-2.5.7, mhash-0.9.7.1, curl-7.15.5, zziplib-0.10.82, bzip2-1.0.3, freetds-0.64

- **Installing Apache in SuSE**

apache2, apache2-doc, apache2-mod_php5, apache2-prefork,
apache2-example-pages (package: libapr0 is also installed as dependency)

- **Start/Stop of Apache** (rcapache2, /etc/init.d/apache2, insserv apache2)

- **First Exercise with HTTP Protocol:** telnet localhost 80 (GET /)

- **Watching the http protocol with Ethereal**

- Install Ethereal from CD/DVD
- Start a browser and Ethereal
- Start Capture of the packets for eth0 and Filter port 80
- Type `http://www.suse.de` in Browser
- Stop Ethereal capture and study packets

- **Configuration files**

Apache 1.3: /etc/apache/httpd.conf

Apache 2.x: /etc/apache2/httpd.conf

or /etc/apache2/apache2.conf

(Include /etc/apache2/sites-enabled/[^.#]*)

also in SuSE only: /etc/sysconfig/apache2

/etc/apache2/

Apache compiled from source code:

/etc/httpd/conf/httpd.conf

Note 1: Directive 'Include *Filename*' extends the configuration to other files.

Important Note for httpd.conf:

NO unrecognized directives allowed otherwise Apache will not start

- **Where is what (basic only)**

- Main server /usr/sbin/httpd2-prefork (Behaves like older Apache 1.3)
or /usr/sbin/httpd2-worker (apache2-mod_php4 doesn't work!)
- Main Configuration file /etc/apache2/httpd.conf
- Modules /usr/lib/apache2/xxxx.so
- Log files /var/log/apache2/xxxx_log
- Main Home pages /srv/www/htdocs/

- **User and group settings for clients processes:**

Directives:

User wwwrun or www-data

Group wwwrun

- **Modules : Roles and their Directives (server-info)**

- Change in /etc/sysconfig/apache2:

APACHE_MODULES="..... status info"

APACHE_EXTENDED_STATUS="on"

- rcaapache2 restart

- In browser: http://localhost/server-info/

- **Apache Server status:** http://localhost/server-status/?refresh=10

- **Documentation:** `http://localhost/manual/`
or `http://www.apache.org`
The local documentation is at `/usr/share/apache2/manual`
(`apache2-doc` package)

- **Important core directives:**

```

In /etc/apache2/default-server.conf
  DocumentRoot /srv/www/htdocs
In /etc/apache2/httpd.conf
  DirectoryIndex index.html
In /etc/apache2/uid.conf
  User wwwrun
  Group www

```

-----Practical Exercises-----

- **Preparation for Exercises:**

- Login as root and create the directory `/www`
- Change in `/etc/sysconfig/apache2`
`APACHE_SERVERNAME="laptop.linux.site"` or local host name
- Create the empty file `/etc/apache2/conf.d/user.conf`
 - Alternative: Change in `/etc/sysconfig/apache2`:
`APACHE_CONF_INCLUDE_FILES="/etc/apache2/conf.d/user.conf"`
and create a empty config file: `/etc/apache2/conf.d/user.conf`
- Transfer and install Opera from Internet and w3m and lynx from CD
- Create Multiple root terminals for:

```

⇒ mcedit /etc/apache2/conf.d/user.conf Title: USER.CONF
⇒ sudo su - (for rcapache2 reload command) Title: RELOAD/RESTART
⇒ sudo su - -c 'tail -f /var/log/apache2/error_log' Title: APACHE FEHLER
⇒ watch -n1 "ps -fC httpd2-prefork" Title: WATCH_APACHE
⇒ sudo su - -c 'mc / /srv/www/htdocs' Title: ROOT-MC
⇒ less /etc/apache2/default-server.conf Title: DEFAULTS
⇒ sudo su - Title: ROOT-BEFEHLE

```

- **Some troubleshooting commands:**

```

netstat -ltupn | grep apache2
apache2 -S
watch 'netstat -ltupn | grep apache2 ; ps ax | grep apache2 | grep -v grep'

```

- **Testing the web server's home pages:**

- Modify in the file `/srv/www/htdocs/index.html.de`
- In browser: `http://localhost/`

ALIAS

Example together:

```

in user.conf: alias /webicons/ /usr/share/apache2/icons/
in browser: http://localhost/webicons/

```

Exercise to do:

in user.conf: alias /software/ /usr/share/doc/packages/
 alias /web/ /www/
 in browser: http://localhost/software/
Result: *ERROR 403 Access forbidden*

in user.conf: <Directory /usr/share/doc/packages>
 Allow from all
 Options +Indexes
</Directory>
<Directory /www>
 Allow from all
 Options +Indexes
</Directory>
 in browser: http://localhost/software/
 http://localhost/web/

Containers

<Directory> Absolute system path of web documents.

<Location> Relative path to DocumentRoot of server (read last after Dir. and Files)

<Files> Can be nested inside <Directory> or alone (Declares defaults for server)

Note: <Files> cannot be nested in <Location>. Use another <Location> for files.
 eg. <Location /usr/local/myprogs/readme.html>

<Limit> Can be anywhere. Used to limit the use of certain HTTP methods

Options FollowSymlinks**Example together:**

In Bash:

ln -s /home/ [username]/Documents /srv/www/htdocs/userdocs

in browser: http://localhost/userdocs/ **Forbidden!**

in user.conf: <Directory /srv/www/htdocs/>
 Options +FollowSymlinks +Indexes
</Directory>

in browser: http://localhost/userdocs/ **Allowed!**

Note: FollowSymlinks Does not work inside a <Location> block

DirectoryIndex**Exercise 1 to do:**

In Bash: ln -s /usr/share/doc/packages/bash /srv/www/htdocs/bash

in user.conf: <Directory /srv/www/htdocs/bash>
 DirectoryIndex bashref.html
</Directory>

in browser: http://localhost/bash/

Exercise 2 to do (if needed):

Note: make sure w3m and lynx are installed

Commands to create the symbolic links:

```
ln -s /usr/share/doc/packages/w3m           /srv/www/htdocs/w3m
ln -s /usr/share/doc/packages/lynx/lynx_help /srv/www/htdocs/lynx
```

```
in user.conf:      <Directory /srv/www/htdocs/w3m>
                  DirectoryIndex w3mhelp.html
                  </Directory>

                  <Directory /srv/www/htdocs/lynx>
                  DirectoryIndex Lynx_users_guide.html
                  </Directory>
```

Allow - Deny**Example1 together (deny access to a single file):**

```
in user.conf:      alias /etc/ /etc/
                  <Directory /etc>
                  Options +Indexes
                  order allow,deny
                  allow from all
                  deny from localhost [dozent_host]
                  <Files fstab>
                  deny from all
                  </Files>
                  <FilesMatch ".*\.conf">
                  order deny,allow
                  deny from all
                  allow from localhost
                  </FilesMatch>
                  </Directory>
```

in browser:	http://localhost/etc/	Allowed!
From Students:	http://localhost/etc/fstab	Not Allowed
	http://localhost/etc/	

Only Dozent is not allowed: http://[studenthost]/etc

Example 2 together:

Allow access to a full directory(location): /cups/ to 4 consecutive addresses

```
in user.conf:      alias /cups/ /usr/share/doc/packages/cups/de/
                  <Location /cups>
                  order deny,allow
                  deny from all
                  allow from 192.168.71.12/30
                  </Location>
```

in browser:	http://localhost/cups/	Not Allowed!
-------------	------------------------	--------------

Only Clients Hosts having the address 192.168.71.12 to 192.168.71.15 are allowed .
to the address: `http://[studenthost]/cups/`

Exercise 1 to do: Allow access to `/sambahelp` only to this class
and have a specific page come up (DirectoryIndex): `welcome.html`

- Make sure the package `samba-doc` package is installed.
- In `user.conf`:


```
alias /sambahelp/ /usr/share/samba/swat/help/
<Directory /usr/share/samba/swat/help/>
    order deny,allow
    deny from all
    allow from 192.168.71.0/27 localhost
    DirectoryIndex welcome.html
</Directory>
```

Only Clients Hosts having the address 192.168.71.0 to 192.168.71.31 are allowed .
to the address: `http://[studenthost]/sambahelp/`

If we click on one of the links we land in a path which doesn't exist.

`http://servername/swat/help/manpages/smbd.8.html`

To enable this link to function we need to create an alias:

`alias /swat/ /usr/share/samba/swat/`

But now the error message is not any more '...NOT FOUND' but FORBIDDEN
the reason is that the `/usr/share/samba/swat/help/manpages` directory
is a symbolic link. Now we need to allow symbolic links to be followed:

```
<Directory /usr/share/samba/swat/help/>
  Options +FollowSymLinks
</Directory>
```

Note: This `Options +FollowSymLinks` can simply be added to the `Directory` container declare previously instead.

Public_html

Note: This function wil only work if the apache module `userdir` is loaded.

Example together:

Install the package `htdig` (if not already installed)

In bash: `cp -R /usr/share/doc/packages/htdig/htdoc/* ~/public_html`

in browser: `http://localhost/~[userlogin]`

Authentication

Create the user : `user1` (`mkdir user1`)

Example together: Authenticate `~/user1/` directory. (`/home/user1/public_html`)

In `user.conf`:

```
<Location ~/user1>
  allow from all
  AuthType Basic
  AuthName "Private Web Page"
  AuthUserFile /etc/apache2/.okusers
```

```

        Require valid-user
    </Location>

in bash: touch /etc/apache2/.okusers
            htpasswd2 /etc/apache2/.okusers user1
in browser: http://localhost/~user1/

```

Exercise 2

```

In user.conf: alias /doc/ /usr/share/doc/
    <Location /doc>
        Options +Indexes
        order deny,allow
        allow from all
        AuthType Basic
        AuthName "Private DOCS"
        AuthUserFile /etc/apache2/.okusers
        # AuthGroupFile /etc/apache2/.okgroups
        Require valid-user
        # Require user jennie otto
        # Require group buero admin
        Satisfy any
        # Satisfy all
    </Location>

in bash: htpasswd2 -c /etc/apache2/.okusers jennie
            htpasswd2 /etc/apache2/.okusers paul
            htpasswd2 /etc/apache2/.okusers marie
            htpasswd2 /etc/apache2/.okusers hans
            htpasswd2 /etc/apache2/.okusers otto
            echo"admin: hans otto"      >> /etc/apache2/.okgroups
            echo"buero: jennie marie"  >> /etc/apache2/.okgroups
            echo"dienst: paul hans"    >> /etc/apache2/.okgroups

in browser: http://localhost/doc/

```

INDEXES

Note: - Index Directives don't work inside `<Location>` container. Use `<Directory>`
 - Alias path can also be controlled by `<Directory>` using the real system Path.
 - Options `FollowSymLinks` cannot be used inside a `<Location>` container.

Example 1 together:

```

Copy the whole directory from the Dozent computer linuxbuch to
                                         /srv/www/htdocs/linuxbuch.

mkdir /srv/www/htdocs/linuxbuch
cp xxxxxxx/linuxbuch/* /srv/www/htdocs/linuxbuch

```

In user.conf:

```

<Directory /srv/www/htdocs/linuxbuch>
    Options +Indexes
    DirectoryIndex dummy
    AddDescription "PDF Datei" .pdf
    AddDescription "<Font color=red>GIF Bilder</Font>" .gif
    AddDescription "<Font color=green>=START=</Font> online.html"

```

----- **Extended Index Features** -----

```

IndexOptions +FancyIndexing \
+FoldersFirst \
+IconsAreLinks \
+SuppressSize \
+SuppressLastModified \
+NameWidth=* \
+DescriptionWidth=* \
+ScanHTMLTitles

HeaderName header.html
ReadmeName footer.html
IndexIgnore header.html footer.html *.txt *.log
</Directory>

```

Note: The '\' at the end of each line above simulate a long line. They should never have anything else than <Enter> after it. No space or tabs or anything.

header.html

```

<HTML>
  <BODY background="green_paper.gif">
    <!-- KOMENTARE SIND HIER -->
    <H1><FONT color=red><CENTER> L I N U X - B U C H <BR>
    Inhalt</CENTER></FONT></H1>
    <IMG src=penguin.gif>
    <H3><FONT color=blue>Letzte Liste von LinuxBu.ch Dateien
    </FONT></H3>
    <BR>
    <HR><HR><HR>
  </BODY>
</HTML>

```

footer.html

```

<B><I>Bitte schreiben Sie Ihre Kommentare durch email:
<a href="mailto:michel@linuxint.com">michel@linuxint.com
</a></I></B>

```

Example 2 together: (Fill-up Descriptions automatically from HTML files titles)
in user.conf:

```

alias      /howto-de/    /usr/share/doc/howto/de/html/
<Directory /usr/share/doc/howto/de/html>
  Options +Indexes
  DirectoryIndex dummy
  IndexOptions +FancyIndexing \
+FolderFirst \
+IconsAreLinks \
+SuppressSize \
+SuppressLastModified \
+NameWidth=* \
+DescriptionWidth=* \
+ScanHTMLTitles
</Directory>

```

in Browser: http://localhost/howto-de/

AllowOverride

Directives: AccessFileName .htaccess

Example 1 together:

(Note: valid only in <Directory> and NOT in <Location>)

In bash:

```
mkdir /srv/www/htdocs/override/
cp /srv/www/htdocs/bash/bashref.html /srv/www/htdocs/override/
```

In browser: http://localhost/override/ File is shown in index

in user.conf:

```
<Directory /srv/www/htdocs/override>
    AllowOverride Indexes
</Directory>
```

In bash: touch /www/override/.htaccess
vi /www/override/.htaccess

In .htaccess: DirectoryIndex bashref.html

In browser: http://localhost/override/ Bash page comes up.

Virtual Hosts

Create the Virtual Host's ROOT directory and its log files Directory:

```
mkdir /www/virtual1.de
mkdir /var/log/apache2/www.virtual1.de/
```

● Enter www.virtual1.de virtual1.de virtual1 in /etc/hosts

eg. echo "\$(hostname -i) www.virtual1.de virtual1.de virtual1" \
>> /etc/hosts

● Fill in the Virtual host with home pages:

```
cp -v /usr/share/doc/packages/rpm/RPM-Tips/* /www/virtual1.de/
```

● in user.conf

```
NameVirtualHost 192.168.71.27
<VirtualHost 192.168.71.27>
    ServerName www.virtual1.de
    ServerAlias virtual1.de virtual1
    DocumentRoot /www/virtual1.de
    TransferLog /var/log/apache2/www.virtual1.de/access_log
    ErrorLog /var/log/apache2/www.virtual1.de/error_log
    <Directory /www/virtual1.de>
        Allow from all
        DirectoryIndex RPM-Tips.html
        # Place extra directives here for this directory
    </Directory>
</VirtualHost>
```

Make sure to set the NoProxy for www.virtual1.de virtual1.de virtual1

Reload or restart Apache: rcapche2 reload or rcapache2 restart

Show the virtual hosts list: httpd -S

Writing scripts for Apache administration(example)

```

#!/bin/bash
# Name:          neue_web_domain
# Zweck: Erzeugt ein neues FTP benutzer und Apache Virtual Host Konfig. Datei
# Syntax:        neue_web_domain
# Author:        michel Bisson (michel@linuxint.com)
# Aenderungen:   12.12.2006 Erste Version
#-----
# ein paar Variable setzen
webdir="/www"
IP="192.168.11.80"

echo \
"Diese Script Erzeugt ein neues FTP benutzer & Apache VirtualHost Konfigdatei"
echo
# ----- Daten eintragen
echo -n "Name der neue Domain eingeben: ";      read domain
echo -n "Name der FTP benutzer eingeben: ";      read benutzer

# ----- Ein paar check machen -----
# ----- ist der Domain schon vorhanden ?
if [ -d $webdir/$domain ]; then
    echo "FEHLER: Domain $domain ist schon vorhanden."
    exit 1
fi
# ----- ist der benutzer schon vorhanden ?
if (grep "^\${benutzer}:" /etc/passwd &>/dev/null); then
    echo "FEHLER: Benutzer $benutzer ist schon vorhanden."
    exit 2
fi
#----- alles ok dann wir machen die Arbeit -----
#----- System vorbereitung -----
mkdir -p $webdir/$domain/htdocs
mkdir -p $webdir/$domain/log
echo "<HTML><BODY><CENTER><H1>Web seite von
Domain<BR>$domain</H1></CENTER></BODY></HTML>" \
> $webdir/$domain/htdocs/index.html
#---- FTP benutzer anlegen-----
useradd -s /bin/false -d $webdir/$domain $benutzer
chown -R $benutzer. $webdir/$domain/
echo "Bitte geben Sie zweimal der Benutzer Passwort"
passwd $benutzer
# ----- Apache VirtualHost einrichtung -----
configdatei="/etc/apache2/vhosts.d/${domain}.conf"

echo "<VirtualHost $IP:80>"                      > $configdatei
echo "    ServerName $domain"                      >> $configdatei
echo "    DocumentRoot $webdir/$domain/htdocs"    >> $configdatei
echo "    <Directory $webdir/$domain/htdocs>"      >> $configdatei
echo "        Allow from all"                      >> $configdatei
echo "        DirectoryIndex index.html"           >> $configdatei
echo "    </Directory>"                           >> $configdatei
echo "    TransferLog $webdir/$domain/log/access_log" >> $configdatei
echo "    ErrorLog $webdir/$domain/log/error_log"    >> $configdatei
echo "</VirtualHost>"                           >> $configdatei

# ----- Der Apache Konfig-Dateien neue lesen -----
/etc/init.d/apache2 reload

# der Apache überprüfen
sleep 2
/etc/init.d/apache2 status

```

Rewrite Module

Allow to redirect an incoming request to another address

Prerequisites: `rewrite` module must be loaded.

For openSuSE: Make sure that the following modules are included the file:

```
/etc/sysconfig/apache2
APACHE_MODULES="ssl rewrite php5 .....
```

Syntax:

Rewrite to a Secure (HTTPS)server

```
RewriteEngine on
RewriteCond %{HTTPS} !^on$ [NC]
RewriteRule . https://www.itmatrix.de/mail [L]
```

or

Rewrite to a Normal HTTP server

```
RewriteEngine on
RewriteCond %{HTTP} !^on$ [NC]
RewriteRule . http://www.itmatrix.de/mail [L]
```

eg.

```
<VirtualHost 192.168.10.35>
    ServerName linuxint.de
    ServerAlias www.linuxint.de
    RewriteEngine on
    RewriteCond %{HTTP} !^on$ [NC]
    RewriteRule . http://www.itmatrix.de/doc [L]
</VirtualHost>
```

CGI

In bash: mkdir /www/cgitest
 echo "192.168.10.50 www.cgitest.net" >> /etc/hosts

In user.conf:

```
NameVirtualHost 192.168.10.50
<VirtualHost 192.168.10.50>
    ServerName www.cgitest.net
    DocumentRoot /www/cgitest
    TransferLog /var/log/apache2/cgitest.access.log
    ErrorLog /var/log/apache2/cgitest.error.log
    <Directory /www/cgitest>
        Allow from all
        Options +execCGI
        AddHandler cgi-script .cgi
        DirectoryIndex cgitest.cgi
    </Directory>
</VirtualHost>

touch /www/cgitest/cgitest.cgi
chmod 755 /www/cgitest/cgitest.cgi
```

cgitest.cgi

```
#!/bin/sh      <----- These 3 following lines MUST be there
echo Content-type: text/html
echo

# This above header and empty echo after it is VERY important.
# Otherwise Apache shows an error : Content Header missing
# The HTML code enclosed in quotes is also very important.....
echo "<HTML><HEAD><TITLE>CGI TEST Seite</TITLE></HEAD>" 
echo "<BODY bgcolor=lightcyan>" 
echo "<H1><CENTER>SYSTEM INFO</CENTER></H1><P>" 
echo "<H3>Ihre IP Addr. ist :</H3><H2>$REMOTE_ADDR</H2><BR>" 

#---Display content of /etc/fstab
echo "<HR><H3>System /etc/fstab</H3><BR><pre>" 
/bin/cat /etc/fstab
echo "</pre><BR><HR>" 

#---Display free space of all mounted disks in Linux -----
echo "<Center><H1>DISK SPACE</H1></Center><BR><pre>" 
/bin/df -h
echo "</pre><HR>" 

#---Display all the system Processes ---
echo "<Center><H3>PROCESS INFO</H3></Center><BR><pre>" 
/bin/ps -eo pid,ppid,user,nice,args --sort user

#---Display Apache's user name and group
echo "</pre><BR><HR>" 
echo "<H3>Apache ist $(id -nu) Benutzer. Er ist Mitglied von
Gruppe $(id -ng)</H3>" 
echo "<BR><HR>" 
echo "<Center><H3>Inhalt von /root Verzeichnis</H3></Center><PRE>" 
/usr/bin/sudo ls /root  (see SUDO below)
echo "</pre><BR><HR>" 
echo "<Center><H3>Inhalt von /etc/shadow Datei</H3></Center><PRE>" 
/bin/cat /etc/shadow      (see SUDO below)
echo "</pre><BR><HR>" 

echo "</BODY></HTML>"
```

SUDO

Edit /etc/sudoers with the command:
visudo

```
wwwrun ALL=NOPASSWD:/usr/bin/sudo ls /root,/bin/cat /etc/shadow
```

----- Verify that the PHP module works -----**1) Create a test php file in current DocumentRoot.**

```
echo "<?php"    >/srv/www/htdocs/phptest.php
echo "<info()" >/srv/www/htdocs/phptest.php
echo "?>"      >/srv/www/htdocs/phptest.php
```

2) In Browser:

```
http://localhost/phptest.php
```

Should give a full home page on the configuration of the PHP module.

----- Creating web access reports with Webalizer -----

In /etc/hosts

```
192.168.10.50 log1.homelinux.net
```

```
#----- LOG INFO of homelinux.net -----
```

```
<VirtualHost 192.168.10.50>
    DocumentRoot /var/log/apache2/homelinux.net/webalized
    ServerName log1.homelinux.net
    <Directory /var/log/apache2/homelinux.net/webalized>
        DirectoryIndex webalize.html
        AuthName LogFiles
        AuthType Basic
        AuthUserFile /etc/apache2/.okusers
        require user michel elop devesh pierre
    </Directory>
    ErrorLog /var/log/apache2/homelinux.net/Log_error.log
    TransferLog /var/log/apache2/homelinux.net/Log_access.log
</VirtualHost>
```

```
-----
```

----/var/log/apache2/homelinux.net/webalize/webalize.html

```
<html>
<head>
    <Title> Web Access Statistics</Title>
</head>
<body>

<center>&nbsp;
<a href="http://log1.homelinux.net/webalize.cgi">Webalize
NOW !!!</a></center>

</body>
</html>
```

```
-----
```

----/var/log/apache2/homelinux.net/webalize/webalize.cgi

```
#!/bin/sh
--- Starts the webalizer for the present log file-----
echo "Content-Type: text/html"
echo ""
sudo /usr/bin/webalizer -Q -n homelinux.net -o . . . /access.log \
```

```
&>/dev/null
cat index.html

-----
---- /etc/sudoers -----
# Host alias specification
root    ALL = (ALL) ALL
wwwrun  ALL = NOPASSWD:/usr/bin/webalizer
```

Secure HTTP: HTTPS configuration:

Make sure that the following modules are included the file:

```
/etc/sysconfig/apache2
APACHE_MODULES="ssl .....
or for SuSE /etc/sysconfig/apache2
APACHE_SERVER_FLAGS="SSL"
```

In user.conf:

```
#----- SSL General Directives -----
Listen 443

SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl
SSLSessionCache dbm:/var/run/apache2/ssl scache (Dependent on system)
SSLMutex file:/var/run/apache2/ssl mutex (Dependent on system)
SSLSessionCacheTimeout 300
SSLCipherSuite \
    ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
#----- SSL Virtual Host -----
<VirtualHost 192.168.10.50:443>
    DocumentRoot /srv/www/htdocs
    SSLEngine on
    SSLCertificateKeyFile /etc/apache2/ssl.key/www.test.com.key
    SSLCertificateFile /etc/apache2/ssl.crt/www.test.com.crt
    <Files ~ "\.(cgi|shtml|phtml|php3?)\$">
        SSLOptions +StdEnvVars
    </Files>

# =====Redirection from www.webmail.com =====
<Directory /srv/www/htdocs/squirrelmail>
    Allow from all
    php_flag register_globals off
    Options Indexes FollowSymLinks
    DirectoryIndex index.php
    <Files configtest.php>
        order deny,allow
        deny from all
```

```

        allow from 127.0.0.1
    </Files>
</Directory>

ErrorLog  /var/log/apache2/https_error_log
CustomLog /var/log/apache2/https_access_log common
</VirtualHost>

```

Creating a temporary self signed certificate for SSL:

3 Methods:

1) Either use the already self-signed temporary certificate files:

/etc/apache2/ssl.crt/snakeoil-dsa.crt

and /etc/apache2/ssl.key/snakeoil-dsa.key

2) Or build your own with the commands:

openssl genrsa 1024 > www.test.com.key

openssl req -new -key www.test.com.key -out www.test.com.csr

openssl req -x509 -key www.test.com.key -in www.test.com.csr -out www.test.com.crt

To test the content of the certificate:

openssl x509 -noout -text -in www.test.com.crt

Copy them both as follows:

cp www.test.com.key /etc/apache2/ssl.key/

cp www.test.com.crt /etc/apache2/ssl.crt/

and use them in both the following directives respectively: eg.

SSLCertificateKeyFile /etc/apache2/ssl.key/www.test.com.key

SSLCertificateFile /etc/apache2/ssl.crt/www.test.com.crt

3) Or best is to produce a xxx.key and xxx.csr files, send it to twarte which will create a certificate (xxx.crt) file from the xxx.key and xxx.csr files.

Secure Webmail Installation (Squirrelmail)

(using the mod_rewrite module)

- Make sure that the following modules are included the file:

```

/etc/sysconfig/apache2
APACHE_MODULES="ssl rewrite php5 .....

```

- In bash:

mkdir /www/

echo "192.168.10.50 webmail.net www.webmail.net" >> /etc/hosts

- Install or copy squirrelmail to /srv/www/htdocs/squirrelmail

- In user.conf

```
# ===== webmail.com =====
```

```
<VirtualHost 192.168.10.50:80>
```

 ServerName webmail.net

 ServerAlias www.webmail.net

 RewriteEngine on

 RewriteCond %{HTTPS} !^on\$ [NC]

 RewriteRule . https://www.webmail.net/squirrelmail/ [L]

```
</VirtualHost>
```

The request will then be redirected to the https VirtualHost configured in the previous section above into a subdirectory(/squirrelmail) where the squirrelmail web page is.

IDS Images Galerie - CGI Installation

1) Install the following Packages from Original SuSE CDs.

```
perl-Image-Info
perl-PerlMagick (Depends on ImageMagick Package)
jpeg (uses jpegtran program)
```

2) Uncompress the the ISD system ids.tar.gz into the directory:

```
/srv/www/htdocs/ids/
```

3) Configure a Virtual Host in Apache2 as follows:

```
<VirtualHost 192.168.100.70>
    ServerName pictures.linux.site
    DocumentRoot /srv/www/htdocs/ids
    <Directory /srv/www/htdocs/ids>
        order deny,allow
        allow from all
        DirectoryIndex index.cgi
        Options +ExecCGI +FollowSymLinks
        AddHandler cgi-script .cgi
        AllowOverride AuthConfig Limit
    </Directory>
</VirtualHost>
```

4) Make sure that owners of the ISD directories are as follows:

```
drwxr-xr-x 11 wwwrun root 456 Apr 27 2002 .
drwxr-xr-x 12 root root 1712 Feb 13 19:44 ..
drwxr-xr-x 3 root root 208 Apr 27 2002 admin
drwxr-xr-x 2 wwwrun root 48 Mar 23 2002 album-data
drwxr-xr-x 2 wwwrun root 48 Mar 23 2002 albums
-rw-r--r-- 1 wwwrun root 737 Mar 23 2002 ids.conf
-rw-r--r-- 1 root root 39426 Dec 16 2001 idsShared.pm
drwxr-xr-x 2 wwwrun root 48 Mar 23 2002 image-cache
-rwxr-xr-x 1 root root 43790 Oct 6 2001 index.cgi
drwxr-xr-x 2 root root 392 Apr 27 2002 localizations
drwxr-xr-x 2 wwwrun root 80 Apr 28 2002 logs
drwxr-xr-x 2 wwwrun root 184 Apr 27 2002 postcomment
-rwxr-xr-x 1 root root 3436 Oct 6 2001 previewmaker.pl
drwxr-xr-x 3 root root 256 Apr 27 2002 site-images
-rw-r--r-- 1 wwwrun root 393 Dec 16 2001 site_news.txt
drwxr-xr-x 9 root root 232 Apr 27 2002 themes
```

5) Create the IDS's administrator password

```
htpasswd -c /etc/apache2/htpasswd.ids admin
```

6) Copy only directories of images into /srv/www/htdocs/ids/albums

It supports also subdirectories, but this root directory must contain only directories.

Note: Make sure that the files and directories copied are owned by wwwrun user.

7) To test the system locally enter the following entry in /etc/hosts:

```
192.168.100.70 pictures.linux.site
```

8) Restart Apache

```
rcaapache2 restart
```

9) In browser enter the address:

`http://pictures.linux.site`

httpd2-prefork command Options

Syntax:	<code>/usr/sbin/httpd2-prefork -options</code>
Options:	
<code>-D name</code>	Defines a name for use in <code><IfDefine name></code> directives <code><IfDefine name></code> is used to define different server global settings and chose which one will be read at start-up of Apache.
<code>-d ServerRootDir</code>	Specifies an alternate initial <code>ServerRoot</code> directory.
<code>-f ConfigFile</code>	Specifies an alternate configuration file.(<code>ServerConfigFile</code>)
<code>-C Directive</code>	Processes this directive <u>before</u> reading config files
<code>-c Directive</code>	Processes this directive <u>after</u> reading config files
<code>-v</code>	Display Apache's version number
<code>-h</code>	List valid command line options
<code>-l (small L)</code>	List compiled-in modules
<code>-L</code>	List core configuration directives
<code>-S</code>	Show virtual hosts settings
<code>-t</code>	Run syntax test for configuration files only.

Controlling the started processes and clients on Apache

```
# prefork MPM (Multi Processing Module)
# StartServers ..... number of server processes to start
# MinSpareServers ..... minimum number of server processes which are kept spare
# MaxSpareServers ..... maximum number of server processes which are kept spare
# MaxClients ..... maximum number of server processes allowed to start
# MaxRequestsPerChild .. maximum number of requests a server process serves
<IfModule prefork.c>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
    MaxClients          20
    MaxRequestsPerChild  0
</IfModule>
```

pthread MPM

```
# StartServers ..... initial number of server processes to start
# MaxClients ..... maximum number of server processes allowed to start
# MinSpareThreads ..... minimum number of worker threads which are kept spare
# MaxSpareThreads ..... maximum number of worker threads which are kept spare
# ThreadsPerChild ..... constant number of worker threads in each server process
# MaxRequestsPerChild .. maximum number of requests a server process serves
<IfModule worker.c>
    StartServers          2
```

```

MaxClients      150
MinSpareThreads 25
MaxSpareThreads 75
ThreadsPerChild 25
MaxRequestsPerChild 0
</IfModule>

```

Apache in Debian based distributions

- Selecting Modules and config files

Extras:

Writing Apache configuration scripts

(Refer to Bash Shell course)

-----Options Multiviews (Not Valid for SuSE 9.1 and 9.2)-----

Optional explanation: Sequence of Indexes delivery:

DirectoryIndex ?	yes: send the <i>file.html</i>
no? then Multiviews ?	yes: send the <i>file.de.html</i> etc.
no? then Options Indexes ?	yes: produce and send index of the directory no: produce an send a 'Forbidden' message

- Check with browser: `http://localhost`
Result: German Web site
- In Mozilla browser: Change the LanguagePriority between DE and EN
- Check with browser: `http://localhost`
Result: English Web site
- in `user.conf`:


```

<Location />
  Options +Multiviews [-Indexes]
</Location>

```
- Comment the above 3 lines after the test is done.

LAMP - Linux-Apache-MySQL-PHP

Install the following packages:

1) Apache2:

```
apache2
apache2-prefork
apache2-mod_php5
```

2) MySQL:

mysql	mysql-client
php5-mysql	knoda

3) phpMyAdmin:

(download) phpMyAdmin	php5-bz2	php5-gd
php5-zlib	php5-mcrypt	pwgen

The above packages are all available on openSuSE 10.2 DVD except for **phpMyAdmin**.

It can be downloaded from the following links.

It is meant for SuSE 9.3 but works just as well in openSuSE 10.2.

(Package: phpMyAdmin-2.9.1.1-2.3.noarch.rpm)

One good version is:

<http://rpmseek.com/rpm/phpMyAdmin-2.9.1.1-2.3.noarch.html?hl=de&cs=phpMyAdmin:PN:0:0:0:0:3400452>

or as tar.gz

http://www.phpmyadmin.net/home_page/downloads.php

These **phpMyAdmin** packages require the following rpm packages which are found on the openSuSE 10.2 DVD:

Configuring phpMyAdmin:

- Copy the file/phpMyAdmin/libraries/config.default.php
to/phpMyAdmin/config.inc.php

eg. cp /srv/www/htdocs//phpMyAdmin/libraries/config.default.php ..

- Edit the file/phpMyAdmin/config.inc.php and do the following changes:

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin';
.....
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```

- Start a browser.

- Allow all cookies to the used in the browser

- Enter the following address: <http://localhost/phpMyAdmin/>

Other ways to use phpMyAdmin:

phpMyAdmin can be called either via a VirtualHost or direct through the main server:

Via VirtualHost

2a) Either create a Virtualhost pointing to /srv/www/htdocs/phpMyAdmin directory

including the Directives:

```
NameVirtualHost 192.168.71.10 (only if doesn't already exist)
<VirtualHost 192.168.71.10>
    ServerName phpmyadmin.linux.site
    DocumentRoot /srv/www/htdocs/phpMyAdmin
    <Directory /srv/www/htdocs/phpMyAdmin>
        Allow from all
        DirectoryIndex index.php
    </Directory>
</VirtualHost>
```

2b) Insert the local IP and server name in /etc/hosts

eg. 129.168.71.10 phpmyadmin.linux.site

Via Main Apache server:

(<http://localhost/phpMyAdmin/>)

4) Start the mysql daemon with the command:

rcmysql start

5) Change the root password in mysql using the commands:

mysqladmin -u root password newpasswd

mysqladmin -p -u root -h localhost password newpasswd

Press <Enter> when asked for the password.

Monitoring MySQL's well working via mysql client:

mysql -p

- Starts the monitor program in terminal
and enter the mysql root password

mysql> show databases;

- Displays all the MySQL Databases

mysql> use mysql;

- Uses the mysql database

mysql> show tables;

- Displays the tables of mysql database

mysql> select * from user \G;

- Displays user table form mysql database

mysql> quit

6) Change the root password and Absolute URI in

/srv/www/htdocs/phpMyAdmin/config.inc.php

eg.

\$cfg['PmaAbsoluteUri'] = 'http://phpmyadmin.linux.site';

or

\$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin';

For Authentication as registered MySQL users then use:

\$cfg['Servers'][\$i]['authtype'] = 'cookie';

7) Start or Restart Apache Server.

rcapache2 restart

8) Start a browser and write the address:

<http://phpmyadmin.linux.site> or <http://localhost/phpMyAdmin/>

9) Protecting the access to phpMyAdmin:

Via VirtualHost

in already configured VirtualHost instructions block:

```
<Directory /srv/www/htdocs/phpMyAdmin>
    Allow from all
    DirectoryIndex index.php
    AuthType Basic
    AuthName "mysql administration"
    AuthUserFile "/etc/apache2/.okmysql"
    require user marie
    satisfy all
</Directory>
```

Via Main Apache server

in any apache configuration file:

```
<Location /phpMyAdmin>
    Allow from all
    DirectoryIndex index.php
    AuthType Basic
    AuthName "mysql administration"
    AuthUserFile "/etc/apache2/.okmysql"
    require user marie
    satisfy all
</Location>
```

Give a password to marie (mysql administrator)

```
touch /etc/apache2/.okmysql
htpasswd2 /etc/apache2/.okmysql marie
```

WAMP-Windows-Apache-MySQL-PHP

Installing Apache2, PHP4 and MySQL in Windows2000/XP

There is a web site where you can download an installer which installs all the components of Apache2, PHP4, MySQL, PhpMyAdmin, Webalizer and extras automatically at: <http://www.apachefriends.org/de/xampp-windows.html>
 The lastest version at the moment of writing was: V. 1.4.11
 Just run the file and all will be done automatically.

Manual installation for the brave!

Note: This installation cuts corners and does things that might not be necessary.
 I've used this method and got a working setup.

For other alternatives, please refer to the document:

<http://www.php.net/manual/en/install.apache2.php>

INSTALLING APACHE2

1) Download the latest stable Apache for Windows from the internet:

<http://www.apache.org>

It's a .msi file and can be installed immediately

2) In my case, I kept the recommended install path of Apache:

C:/programm(e)/Apache Group/Apache2/

INSTALLING MYSQL

1) Download the MySQL windows installer from the Internet:

<http://dev.mysql.com/downloads/mysql/4.0.html>

It's a ZIP file

2) Unpack the ZIP file into a temporary directory and run the program SETUP.EXE

I installed it in C:/mysql directory

INSTALLING PHP4(as apache module)

1) Download the ZIP file from:

<http://www.php.net/downloads.php>

2) Unzip the files into C:/php directory

3) copy all the .dll files from C:/php/dlls and C:/php/sapi to C:/php directory

INSTALLING phpMyAdmin

1) Download the latest version of phpMyAdmin ZIP file from:

http://www.phpmyadmin.net/home_page/

2) Unzip the file into the directory:

C:/Programme/Apache Group/Apache2/htdocs/phpmyadmin\

3) Edit the config.inc.php in the phpmyadmin directory and:

- add the proper full URI of the phpMyAdmin:

\$cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin/';

Optional:(If you leave the root password blank in MySQL then leave it blank here also)

- add the MySQL root (administrator) password in clear text in:

```
$cfg['Servers'][$i]['user']      = 'root'; // MySQL user
$cfg['Servers'][$i]['password']  = '';      // MySQL password
                                         (only needed with 'config' auth_type)
```

CONFIGURATION

1) Edit the `httpd.conf` file and:

- Change to the following settings to:

```
DocumentRoot "C:/Programme/Apache Group/Apache2/htdocs"  
DirectoryIndex index.html index.html.var index.php
```

- Add the following settings:

```
LoadModule php4_module c:/php/php4apache2.dll  
LoadFile C:/php/php4ts.dll  
AddType application/x-httpd-php .php .php3 .php4 .phtml
```

2) Edit the `C:\php\php.ini` file

- You will need to change the `extension_dir` setting to point to your `php-install-dir`, or where you have placed your '`php_*.dll`' files. eg: `c:\php`
It is located around the line 442
- Set the '`doc_root`' to point to your webserver's `document_root`.
in my case: `C:/Programme/Apache Group/Apache2/htdocs`

3) Create the file `C:\MY.CNF` and insert the following lines in it:

```
[mysqld]  
basedir=C:/mysql/  
datadir=C:/mysql/data/
```

Note: See the `MY.CNF` example from XAMPP below

OPERATING WAMP

1) Open a DOS window (`cmd`) and give the command:

```
C:\mysql\bin\mysqld.exe
```

When all is ok then an icon can be created to start it either from the autostart area or by hand.

2) start Apache through the Windows start menu

3) Start a browser and give the address:

```
http://localhost/phpmyadmin/
```

You should have a `phpMyAdmin` web site with full control over the MySQL databases. This would confirm that your Apache, PHP and MySQL are all working...so far go good!

MY.CNF (Example from XAMPP)

```

# The MySQL client
[client]
password      = your_password
port          = 3306
socket        = c:/WAMP/xampp/mysql/mysql.sock

# Here follows entries for some specific programs

# The MySQL server
[mysqld]
port          = 3306
socket        = c:/WAMP/xampp/mysql/mysql.sock
skip-locking
set-variable  = key_buffer=16M
set-variable  = max_allowed_packet=1M
set-variable  = table_cache=64
set-variable  = sort_buffer=512K
set-variable  = net_buffer_length=8K
set-variable  = myisam_sort_buffer_size=8M
log-bin
server-id    = 1

basedir=c:/WAMP/xampp/mysql
tmpdir=c:/WAMP/xampp/tmp
datadir=c:/WAMP/xampp/mysql/data

#bind-address=192.168.1.1
#log-update      = /path-to-dedicated-directory/hostname

# Uncomment the following if you are using BDB tables
#set-variable  = bdb_cache_size=4M
#set-variable  = bdb_max_lock=10000

skip-innodb

# Uncomment the following if you are using InnoDB tables
#innodb_data_home_dir = c:/WAMP/xampp/mysql/
#innodb_data_file_path = ibdata1:10M:autoextend
#innodb_log_group_home_dir = c:/WAMP/xampp/mysql/
#innodb_log_arch_dir = c:/WAMP/xampp/mysql/
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
#set-variable = innodb_buffer_pool_size=16M
#set-variable = innodb_additional_mem_pool_size=2M
# Set .._log_file_size to 25 % of buffer pool size
#set-variable = innodb_log_file_size=5M
#set-variable = innodb_log_buffer_size=8M
#innodb_flush_log_at_trx_commit=1
#set-variable = innodb_lock_wait_timeout=50

[mysqldump]

```

```
quick
set-variable      = max_allowed_packet=16M

[mysql]
no-auto-rehash
# Remove the next comment character if you are not familiar with SQL
#safe-updates
[isamchk]
set-variable      = key_buffer=20M
set-variable      = sort_buffer=20M
set-variable      = read_buffer=2M
set-variable      = write_buffer=2M

[myisamchk]
set-variable      = key_buffer=20M
set-variable      = sort_buffer=20M
set-variable      = read_buffer=2M
set-variable      = write_buffer=2M

[mysqlhotcopy]
interactive-timeout

[WinMySQLadmin]
Server=c:/WAMP/xampp/mysql/bin/mysql.exe
```